

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1. (Currently Amended) A computer-implemented method of reducing redundancy within a data model in a database, wherein the data model is represented by at least one table, the method comprising:

determining a number of distinct values of partial keys in a table, wherein each partial key represents at least one row in the table;

reordering one or more columns of the table by cardinality of partial keys, wherein the cardinality of a partial key represents a number of distinct values of the partial key;

determining whether pairs of partial keys are functionally dependent, each of the pairs comprising a first partial key and a second partial key; and

eliminating one or more columns having functional dependencies from the table.

2. (Currently Amended) The method in accordance with claim [[Claim]] 1, further comprising:

placing the one or more eliminated columns into a separate table so that the column with a highest cardinality is in the leftmost position, and the column with the lowest cardinality is in the rightmost position.

3. (Currently Amended) The method in accordance with claim [[Claim]] 2, wherein partial key K(i) comprises a partial key with an index i and a value Kri for a tuple t(r) in

row with index  $r$ , and wherein the number of distinct values of  $K(i)$  comprises cardinality  $|K(i)|$ .

4. (Currently Amended) The method in accordance with claim [[Claim]] 3, wherein the tuple  $t$  comprises  $k$  key figures and  $d$  partial keys  $K(1), \dots, K(d)$ .
5. (Currently Amended) The method in accordance with claim [[Claim]] 4, wherein a table  $T$  comprises  $n$  tuples and  $d+k$  columns, wherein the  $n$  tuples comprise rows.
6. (Currently Amended) The method in accordance with claim [[Claim]] 5, wherein a function  $F(x) = y$  comprises a mapping between partial keys  $x$  and  $y$  in a same tuple.
7. (Currently Amended) The method in accordance with claim [[Claim]] 6, wherein a flag  $fd$  comprises Boolean values of true or false indicative of whether a value of the function  $F$  is defined.
8. (Currently Amended) The method in accordance with claim [[Claim]] 7, wherein determining whether pairs of partial keys are functionally dependent further comprises:
  - defining function  $F$  from each partial key to every other partial key to its right in a reordered table for each row in table  $T$ ; and
  - determining a functional dependency exists when the function  $F(Kri) = Krj$  is the same function for each tuple  $t(r)$  in the table for values of index  $i$  from 1 to  $(d - 1)$  and for values of  $j$  from  $(i + 1)$  to  $d$ .

9. (Currently Amended) The method in accordance with claim [[Claim]] 8, wherein when a tuple  $t$  is in the table  $T$  and  $F(K_{ri})$  is not equal to  $K_{rj}$ , a functional dependency does not exist between columns  $i$  and  $j$ .

10. (Currently Amended) The method in accordance with claim [[Claim]] 9, wherein determining whether pairs of partial keys are functionally dependent for each  $i$  from 1 to  $(d - 1)$  and  $j$  from  $(i + 1)$  to  $d$  comprises:

setting the flag  $fd$  to true;

for each tuple  $t$  in  $T$ , determining whether  $F(K_{ri})$  is defined, wherein  $F(K_{ri})$  is set equal to  $K_{rj}$  upon determining that  $F(K_{ri})$  is not defined;

looping through the tuples  $t$  in  $T$ ; and

generating a report indicating that column  $i$  is functionally dependent on column  $j$  if flag  $fd$  is true after the looping through the tuples  $t$  in  $T$ .

11. (Currently Amended) The method in accordance with claim [[Claim]] 10, wherein determining whether  $F(K_{ri})$  is defined comprises:

upon determining that  $F(K_{ri})$  is defined, determining whether  $F(K_{ri})$  is equal to  $K_{rj}$ , wherein determining that  $F(K_{ri})$  is equal to  $K_{rj}$  permits looping through the tuples  $t$  in  $T$ , and wherein determining that  $F(K_{ri})$  is not equal to  $K_{rj}$  comprises:

concluding that  $K(i)$  is not functionally dependent on  $K(j)$ ;

setting flag  $fd$  to false; and

breaking the looping through the tuples  $t$  in  $T$ .

12. (Original) A computer-implemented method of reducing redundancy within a data model in a database, wherein the data model is represented by at least one table, the method comprising:

determining a number of distinct values of partial keys in a table, wherein each partial key represents at least one row in the table;

reordering one or more columns of the table by cardinality of partial keys, wherein the cardinality of a partial key represents a number of distinct values of the partial key;

determining whether pairs of partial keys are functionally dependent;

eliminating one or more columns having functional dependencies from the table;

and

creating an exception list for the pairs of partial keys that are not functionally dependent.

13. (Currently Amended) The method in accordance with claim [[Claim]] 12, wherein partial key  $K(i)$  comprises a partial key with an index  $i$  for a tuple  $t$ , wherein the number of distinct values of  $K(i)$  comprises cardinality  $|K(i)|$ , wherein the tuple  $t$  comprises  $k$  key figures and  $d$  partial keys  $K(i)$  for  $i$  from 1 to  $d$ , wherein a table  $T$  comprises  $n$  tuples and  $(d + k)$  columns, wherein the  $n$  tuples comprise rows, and wherein a function  $F(x) = y$  comprises a mapping between partial keys  $x$  and  $y$  in a same tuple.

14. (Currently Amended) The method in accordance with claim [[Claim]] 12, wherein the exception list for the pairs of partial keys that are not functionally dependent comprises partial keys pairs that do not fit a functional dependency defined for other

tuples in the table, and wherein the exception list represents errors in the one or more data models.

15. (Currently Amended) The method in accordance with claim [[Claim]] 14, wherein the determining whether pairs of partial keys are functionally dependent comprises:

defining function  $F$  from each partial key to every other partial key to its right in a reordered table for each row in table  $T_i$  and

determining a functional dependency exists for  $i$  from 1 to  $(d - 1)$  and  $j$  from  $(i + 1)$  to  $d$ , wherein the function  $F(K_{ri}) = K_{rj}$  is the same function for each tuple  $t(r)$  in the table.

16. (Currently Amended) The method in accordance with claim [[Claim]] 15 wherein when for  $i$  from 1 to  $(d - 1)$  and  $j$  from  $(i + 1)$  to  $d$  the function  $F(K_{ri}) = K_{rj}$  is not the same function for each tuple  $t(r)$  in the table, there exists one or more mappings from  $K_{ri}$  to  $K_{rj}$  for different values of  $r$ , wherein different values of  $r$  are related to different tuples  $t(r)$ , and upon determining multiple mappings, checking whether one or more entries in set  $\{K_{rj}\}$  are similar for each  $t(r)$ .

17. (Currently Amended) The method in accordance with claim [[Claim]] 16, wherein a similarity is defined with any one of a similarity function and a data cleansing function.

18. (Currently Amended) The method in accordance with claim [[Claim]] 17, wherein if a subset of  $\{K_{rj}\}$  is similar, compress the subset to a single value  $x$  to compress multiple mappings to a single functional dependency, wherein if a subset of  $\{K_{rj}\}$  is not similar, create an exception list for non-similarities, and wherein the creating an

exception list for non-similarities comprises mapping a row number  $r$  for tuple  $t(r)$  of each dissimilar entry  $Krj$  to a corresponding value  $Kri$ .

19. (Currently Amended) The method in accordance with claim [[Claim]] 18, further comprising rewriting one or more queries against the table to check the exception list before accessing function  $F$ , wherein if no entry exists for the current row in that list, use the functional dependency defined by  $F$ .

20. (Original) An article comprising a machine-readable medium storing instructions operable to cause a machine to perform operations comprising:

reducing redundancy within a data model in a database, wherein the data model is represented by at least one table, the reducing redundancy comprising:

determining a number of distinct values of partial keys in a table, wherein each partial key represents at least one row in the table;

reordering one or more columns of the table by cardinality of partial keys, wherein the cardinality of a partial key represents a number of distinct values of the partial key;

determining whether pairs of partial keys are functionally dependent; and

eliminating one or more columns having functional dependencies from the table.